

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Takanori TAMAI

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HEREWITH

FOR: INTEGRATED CIRCUIT DESIGN SYSTEM AND METHOD USING PREPROCESSOR WHICH
CHANGES HARDWARE DESCRIPTION IN ACCORDANCE WITH CONFIGURATION

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

☐ Full benefit of the filing date of U.S. Application Serial Number _____, filed _____, is claimed pursuant to the provisions of **35 U.S.C. §120**.

☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of **35 U.S.C. §119(e)**: Application No. _____ Date Filed _____

☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of **35 U.S.C. §119**, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

COUNTRY

APPLICATION NUMBER

MONTH/DAY/YEAR

Japan

2003-096687

March 31, 2003

Certified copies of the corresponding Convention Application(s)

☒ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

☐ were filed in prior application Serial No. _____ filed _____

☐ were submitted to the International Bureau in PCT Application Number _____
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

☐ (A) Application Serial No.(s) were filed in prior application Serial No. _____ filed _____; and

☐ (B) Application Serial No.(s) _____

☐ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124



22850

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 3月31日

出 願 番 号

Application Number:

特願2003-096687

[ST.10/C]:

[JP2003-096687]

出 願 人

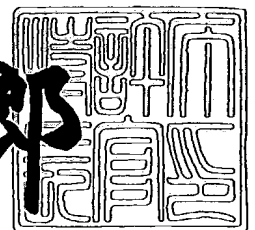
Applicant(s):

株式会社東芝

2003年 4月18日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



【書類名】 特許願

【整理番号】 A000202777

【提出日】 平成15年 3月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/50

【発明の名称】 プリプロセッサ、集積回路の設計システム及び集積回路
の設計方法

【請求項の数】 20

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝マ
イクロエレクトロニクスセンター内

【氏名】 玉井 孝典

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100108855

【弁理士】

【氏名又は名称】 蔵田 昌俊

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 プリプロセッサ、集積回路の設計システム及び集積回路の設計方法

【特許請求の範囲】

【請求項1】 プリプロセッサ制御ファイルに基づいて、第1のハードウェア記述言語と第2のハードウェア記述言語が混在する第1の回路記述ファイルの処理を行い、前記第1の回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換して第2の回路記述ファイルを生成し、出力することを特徴とするプリプロセッサ。

【請求項2】 前記プリプロセッサ制御ファイルと前記第1の回路記述ファイルとに基づいて、ゲートドクロック回路用の論理合成制御スクリプトファイルを更に生成して出力することを特徴とする請求項1に記載のプリプロセッサ。

【請求項3】 前記第2のハードウェア記述言語は、Verilog-HDLまたはVHDLを含むことを特徴とする請求項1または2に記載のプリプロセッサ。

【請求項4】 前記第1の回路記述ファイルの処理は、前記第1の回路記述ファイルから前記第1のハードウェア記述言語を抽出して前記第2のハードウェア記述言語に変換し、前記第1の回路記述ファイル中の前記第2のハードウェア記述言語で記述された部分は変換せずに出力するものであることを特徴とする請求項1乃至3いずれか1つの項に記載のプリプロセッサ。

【請求項5】 前記第1の回路記述ファイルは、第1のハードウェア記述言語で記述されたフリップフロップに関する記述を含み、前記フリップフロップの記述をゲートドクロック化した回路に対応する情報を第2のハードウェア記述言語に変換して生成し、出力することを特徴とする請求項1乃至4いずれか1つの項に記載のプリプロセッサ。

【請求項6】 前記第1のハードウェア記述言語で記述されたフリップフロップに関する記述は、フリップフロップのリセット方式が定まっていない状態で記述され、前記第1の回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換する際に、前記

フリップフロップが同期リセット方式か非同期リセット方式かを指定することを特徴とする請求項5に記載のプリプロセッサ。

【請求項7】 前記フリップフロップに関する記述は、1つのフリップフロップ記述に1つのクラスタ番号が付与されていることを特徴とする請求項5または6に記載のプリプロセッサ。

【請求項 8】 前記第 2 の回路記述ファイルは、複数のクラスタ番号で示されたフリップフロップの記述が、1 つのゲーティングされたクロック信号で駆動されるフリップフロップの記述に変換されたものであることを特徴とする請求項 5 乃至 7 いずれか 1 つの項に記載のプリプロセッサ。

【請求項 9】 プリプロセッサ制御ファイルに基づいて、第 1 のハードウェア記述言語で記述されたフリップフロップの記述を含む第 1 の回路記述ファイルの処理を行い、少なくとも前記フリップフロップの記述を第 2 のハードウェア記述言語に変換して第 2 の回路記述ファイルを生成し、且つゲेटッドクロック回路用の論理合成制御スクリプトファイルを生成するプリプロセッサと、

前記プリプロセッサで生成された前記第2の回路記述ファイルと前記論理合成制御スクリプトファイルとを論理合成して、回路を構成する基本単位であるセルを使用した回路記述ファイルに変換することによりネットリストを生成し、前記ネットリストに基づいて集積回路におけるセルと配線の配置を決定する論理合成ツールと

を具備することを特徴とする集積回路の設計システム。

【請求項 10】 前記ネットリストは、前記第 2 の回路記述ファイルと前記論理合成制御スクリプトファイルに加えて、更にゲートッドクロック以外の制御を行うための論理合成制御スクリプトファイルを論理合成して生成されることを特徴とする請求項 9 に記載の集積回路の設計システム。

【請求項 11】 前記論理合成ツールによる論理合成によって、フリップフロップのゲートドクロック化のための情報を生成することを特徴とする請求項 9 または 10 に記載の集積回路の設計システム。

【請求項 12】 前記第 1 のハードウェア記述言語で記述されたフリップフロップに関する記述は、フリップフロップのリセット方式が定まっていない状態

で記述され、前記第1の回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換する際に、前記フリップフロップが同期リセット方式か非同期リセット方式かが指定されることを特徴とする請求項9乃至11いずれか1つの項に記載の集積回路の設計システム。

【請求項13】 前記フリップフロップに関する記述は、1つのフリップフロップ記述に1つのクラスタ番号が付与されることを特徴とする請求項9乃至12いずれか1つの項に記載の集積回路の設計システム。

【請求項14】 前記第2の回路記述ファイルは、複数のクラスタ番号で示されるフリップフロップの記述が、1つのゲーティングされたクロック信号で駆動されるフリップフロップの記述に変換されたものであることを特徴とする請求項9乃至12いずれか1つの項に記載の集積回路の設計システム。

【請求項15】 第1のハードウェア記述言語と第2のハードウェア記述言語が混在する回路記述ファイルと、プリプロセッサの動作を制御するためのプリプロセッサ制御ファイルとをプリプロセッサに入力し、前記回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換するステップと、

前記プリプロセッサから出力される回路記述ファイルとゲーテッドクロック回路用の論理合成制御スクリプトファイル、及びゲーテッドクロック回路以外の論理合成制御スクリプトファイルを論理合成ツールで論理合成し、回路を構成する基本単位であるセルを使用した回路記述ファイルに変換してネットリストを生成するステップと、

前記ネットリストに基づいて前記セルと配線の配置を決定し、チップの回路設計を行うステップと

を具備することを特徴とする集積回路の設計方法。

【請求項16】 前記回路記述ファイルは、第1のハードウェア記述言語で記述されたフリップフロップの記述を含み、前記プリプロセッサによって同期／非同期指定とゲーテッドクロックのクラスタ結合が変更されることを特徴とする請求項15に記載の集積回路の設計方法。

【請求項 17】 前記プリプロセッサ制御ファイルは、リセット信号の名前、クロック信号の名前、フリップフロップのリセット方式、ゲートッドクロック化するか否か、ゲートッドクロック化する場合に、そのクロックゲーティングの記述をプリプロセッサが生成するか否か、論理合成プログラムの自動ゲートッドクロック化機能が自動認識できる記述形式で出力するか否か、及びどのクラスタを 1 つと見なすかに関する情報の少なくともいずれか 1 つを含むことを特徴とする請求項 16 に記載の集積回路の設計方法。

【請求項 18】 前記回路記述ファイルにおける第 1 のハードウェア記述言語で記述された少なくとも一部を第 2 のハードウェア記述言語に変換するステップは、

前記プリプロセッサ制御ファイルを読み込んで解釈し、この解釈した情報を前記プリプロセッサに格納するステップと、

前記プリプロセッサに格納した情報が拡張記述に関するか否か判定し、拡張記述であったときに拡張記述の解析を行うステップと、

拡張記述でないと判定されたときにタグの判定を行うステップと、

所定のタグであると判定されたときに、クラスタ結合指定の判定を行うステップと、

クラスタ結合指定がないときに、フリップフロップを同期リセットにするか非同期リセットにするかを指定し、同期リセットのフリップフロップに対応する回路記述または非同期リセットのフリップフロップに対応する回路記述を生成するステップと、

クラスタ結合指定があるときに、クラスタ結合に必要な情報を生成するステップと

を備えることを特徴とする請求項 15 乃至 17 いずれか 1 つの項に記載の集積回路の設計方法。

【請求項 19】 前記拡張記述の解析を行うステップは、

フリップフロップに関する拡張記述か否かを判定するステップと、

前記判定ステップでフリップフロップに関する拡張記述であると判定されたときに、構造体フリップフロップの各変数へ情報を格納するステップと、

前記構造体フリップフロップの実体へのポインタに対してタグを付加し、一時バッファに格納するステップと

を備えることを特徴とする請求項 1 8 に記載の集積回路の設計方法。

【請求項 2 0】 前記タグの判定を行うステップは、

前記一時バッファからタグ及び格納情報を取得するステップと、

前記タグが所定のタグか否か判定し、所定のタグではないときに前記格納情報をそのまま出力ファイルへ出力するステップと

を備えることを特徴とする請求項 1 8 に記載の集積回路の設計方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

この発明は、ハードウェア記述をコンフィギュレーションに従って変更する作業を自動化する場合などにおいて使用されるプリプロセッサ、このプリプロセッサを用いた集積回路の設計システム、及びこのシステムによる集積回路の設計方法に関する。特に I P, ソフト I P をユーザの指定に従って自動的に生成するシステムなどに好適なもので、フリップフロップの同期／非同期指定やゲーテッドクロックのクラスタ結合を変更可能な回路記述ファイルを生成するためのものである。

【 0 0 0 2 】

【従来の技術】

従来、集積回路の設計システムは、例えば図 1 3 に示すように構成されている。このような集積回路を設計するためのコンピュータシステムとその方法については、例えば特許文献 1 に記載されている。

【 0 0 0 3 】

ユーザによって既存の言語（V e r i l o g - H D L : Verilog-Hardware Description Language または V H D L : Very High Speed Integrated Circuits Hardware Description Language など）で記述された回路記述ファイル 1 1 と、論理合成制御スクリプトファイル（論理合成プログラムを制御するためのスクリプトファイル） 1 2 がプロセッサ（コンピュータ） 1 3 の論理合成プログラム 1 4 によ

って論理合成される。論理合成プログラム14では、上記回路記述ファイル11を、回路を構成する基本単位である「セル」を使用した回路記述ファイルに変換する処理が行われ、ネットリスト (Netlist) 15が生成される。このネットリスト15が配置配線プログラム (セル及び配線を配置するプログラム) 16により処理されて、セルと配線の配置が決定され、チップ (半導体集積回路) 17の回路設計が行われる。

【0004】

ところで、上記のような従来の集積回路の設計システムでは、Verilog-HDLやVHDLなどの既存の言語を使用して同期リセット回路や非同期リセット回路を記述する場合には、同期リセット用と非同期リセット用の2つの記述を手で作成しなければならない。また、低消費電力化のためにフリップフロップのゲーテッドクロック化を行う場合、何個のフリップフロップを1つのクラスタとするのが最適であるか、あるいはどのフリップフロップとどのフリップフロップがチップ中で近くに配置されているからクラスタ結合をすべきか、などの条件が種々の要因で異なるため最適なクラスタの構成方法が異なってくる。上記要因としては、例えば下記(1)～(3)がある。

【0005】

- (1) 半導体集積回路 (ICやLSI) の物理的なテクノロジーの相違。

【0006】

- (2) 回路の動作パターン、すなわちユーザアプリケーションがその回路をどのように使用するかの違い。

【0007】

- (3) チップ中での各回路の配置の仕方。

【0008】

しかし、従来は人手により、同期リセット用と非同期リセット用の記述を作成したり、ゲーテッドクロック化のための回路を試行錯誤で挿入していたので、最適値に収束させるのが難しい。しかも、ゲーテッドクロック化のための回路を挿入する場合には周辺の回路を修正する必要があり、フリップフロップの記述周辺の修正だけとはいえ、回路規模が大きいと数千ヶ所になることもある。また、人

手による修正では、バグの混入を避けることができず、修正のたびに機能検証作業が必要となる。LSIの回路規模や複雑度は大変な速度で上昇しており、これに伴って検証作業にかかる時間も膨大になっている。このため、開発期間の長期化やコストの上昇の1つの要因になっている。

【0009】

なお、上記ゲートドクロック化のための回路を自動的に挿入するツールは既に存在しているが（例えば非特許文献1参照）、指定したクラスタを結合するなどの細かな要求に応えることはできず、必ずしも満足できるものではない。

【0010】

【特許文献1】

U.S. Patent No. 5,987,239. Graham Kirsch "COMPUTER SYSTEM AND METHOD FOR BUILDING A HARDWARE DESCRIPTION LANGUAGE REPRESENTATION OF CONTROL LOGIC FOR A COMPLEX DIGITAL SYSTEM" Nov. 16, 1996.

【0011】

【非特許文献1】

0-8186-4350-1/93 1993 IEEE pp.466-471 "A Workbench for Generation of Component Models" Marcus Bluml et al.

【0012】

【発明が解決しようとする課題】

上述したように、従来のプリプロセッサ、集積回路の設計システム及び集積回路の設計方法では、人手による作業が必要なため、時間がかかりバグの混入も不可避であり、開発期間の長期化やコストの上昇の要因となる、という問題があった。

【0013】

また、ゲートドクロック化のための回路を自動的に挿入するツールは存在するが、細かな要求に応えることができない、という問題があった。

【0014】

この発明は上記のような事情に鑑みてなされたもので、その目的とするところは、フリップフロップの同期／非同期指定及びゲートドクロックのクラスタ結

合を変更可能な回路記述ファイルを生成できるプリプロセッサを提供することにある。

【0015】

また、開発期間を短縮し、コストを低減できる集積回路の設計システムとこのシステムによる集積回路の設計方法を提供することにある。

【0016】

【課題を解決するための手段】

この発明の一態様によると、プリプロセッサ制御ファイルに基づいて、第1のハードウェア記述言語と第2のハードウェア記述言語が混在する第1の回路記述ファイルの処理を行い、前記第1の回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換して第2の回路記述ファイルを生成し、出力するプリプロセッサが提供される。

【0017】

上記のような構成のプリプロセッサを用いれば、人手を介在することなく、フリップフロップの同期／非同期指定やゲーテッドクロックのクラスタ結合を変更可能な回路記述ファイル（第2の回路記述ファイル）を生成できる。よって、人手による作業時間を短縮するとともに、バグの混入を避けることができ、検証作業にかかる時間も不要になる。

【0018】

また、この発明の一態様によると、プリプロセッサ制御ファイルに基づいて、第1のハードウェア記述言語で記述されたフリップフロップの記述を含む第1の回路記述ファイルの処理を行い、少なくとも前記フリップフロップの記述を第2のハードウェア記述言語に変換して第2の回路記述ファイルを生成し、且つゲーテッドクロック回路用の論理合成制御スクリプトファイルを生成するプリプロセッサと、前記プリプロセッサで生成された前記第2の回路記述ファイルと前記論理合成制御スクリプトファイルとを論理合成して、回路を構成する基本単位であるセルを使用した回路記述ファイルに変換することによりネットリストを生成し、前記ネットリストに基づいて集積回路におけるセルと配線の配置を決定する論理合成ツールとを具備する集積回路の設計システムが提供される。

【0019】

上記のような構成のシステムによれば、プリプロセッサでフリップフロップの同期／非同期指定やゲートドクロックのクラスタ結合を変更可能な回路記述ファイル（第2の回路記述ファイル）を生成するので、従来は人手による作業で行っていた部分を自動化して集積回路を設計できる。これによって、作業時間を短縮し、且つ人手の介在によるバグの混入を避けることができ、検証作業にかかる時間も不要になる。よって、開発期間を短縮し、コストを低減できる。

【0020】

更に、この発明の一態様によると、第1のハードウェア記述言語と第2のハードウェア記述言語が混在する回路記述ファイルと、プリプロセッサの動作を制御するためのプリプロセッサ制御ファイルとをプリプロセッサに入力し、前記回路記述ファイルにおける第1のハードウェア記述言語で記述された少なくとも一部を第2のハードウェア記述言語に変換するステップと、前記プリプロセッサから出力される回路記述ファイルとゲートドクロック回路用の論理合成制御スクリプトファイル、及びゲートドクロック回路以外の論理合成制御スクリプトファイルを論理合成ツールで論理合成し、回路を構成する基本単位であるセルを使用した回路記述ファイルに変換してネットリストを生成するステップと、前記ネットリストに基づいて前記セルと配線の配置を決定し、チップの回路設計を行うステップとを具備する集積回路の設計方法が提供される。

【0021】

上記のような方法によれば、従来は人手による作業で行っていた部分を自動化して集積回路を設計できる。これによって、作業時間を短縮し、且つ人手の介在によるバグの混入を避けることができ、検証作業にかかる時間も不要になる。この結果、開発期間を短縮し、コストを低減できる。

【0022】

【発明の実施の形態】

以下、この発明の実施の形態について図面を参照して説明する。

図1は、この発明の実施の形態に係るプリプロセッサ及び集積回路の設計システムについて説明するためのもので、システム全体の概略構成を示している。ユ

ーザによって記述された回路記述ファイル 21 と、同じくユーザによって記述されたプリプロセッサ制御ファイル（プリプロセッサの動作を制御するための記述ファイル）22 がプリプロセッサ 23 に入力され、プリプロセッサ制御ファイル 22 の制御データに基づいて回路記述ファイル 21 に対する処理が行われる。上記回路記述ファイル 21 には、異なるハードウェア記述言語（ここでは記述言語 A と記述言語 B と称する）が混在している。また、上記回路記述ファイル 21 には、下記（a）～（f）の情報が記述されている。

【0023】

- （a）リセット信号の名前
- （b）クロック信号の名前
- （c）フリップフロップのリセット方式 [同期／非同期]
- （d）ゲートッドクロック化するか否か [YES／NO]
- （e）ゲートッドクロック化する場合に、そのクロックゲーティングの記述をプリプロセッサが生成するか否か

論理合成プログラムの自動ゲートッドクロック化機能が自動認識できる記述形式で出力するか否か [自動／プリプロセッサが生成]

- （f）どのクラスタを 1 つと見なすか（結合）に関する情報（複数可）

上記プリプロセッサ 23 により、上記回路記述ファイル 21 中の記述言語 A が既存の記述言語 B（Verilog-HDL または VHDL など）に変換され、回路記述ファイル 24 が作成される。また、論理合成プログラム 25 でゲートッドクロック化を行う際に使用する論理合成制御スクリプトファイル（ゲートッドクロック回路用）26 が作成される。

【0024】

論理合成ツール（コンピュータ）27 には、上記回路記述ファイル 24 及び上記論理合成制御スクリプトファイル 26 に加えて、予め用意されているゲートッドクロックに関するもの以外の論理合成スクリプトファイル 28 が入力される。論理合成プログラム 25 では、上記回路記述ファイル 24 を、回路を構成する基本単位であるセルを使用した回路記述ファイルに変換する処理が行われる。この段階では、セル及び配線をどのように配置するかは決まっていない。そして、こ

の論理合成プログラム25によってネットリスト（論理合成プログラムから出力されたセルを使用した回路記述）29が生成される。

【0025】

上記ネットリスト29は、配置配線プログラム（セル及び配線を配置するプログラム）30により処理されてセルと配線の配置が決定される。そして、この配置配線プログラム30の処理結果に基づいて、チップ（半導体集積回路）31の回路設計が行われる。

【0026】

図2乃至図7はそれぞれ、上記図1に示した集積回路の設計システムにおけるプリプロセッサ23の動作を示すフローチャートである。図2は図1に示したシステムにおけるプリプロセッサの概略的な動作を示すフローチャート、図3は第1ステージの処理サブルーチンを示すフローチャート、図4は拡張記述解析サブルーチンを示すフローチャート、図5は第2ステージの処理サブルーチンを示すフローチャート、図6はタグBの処理サブルーチンを示すフローチャート、図7はクラスタ結合サブルーチンを示すフローチャートである。

【0027】

図2に示すように、まず、プリプロセッサ23のプログラム本体の動作が開始されると、プリプロセッサ制御ファイルの読み込みとその解釈が行われ、その情報が配列CTRLへ格納される（STEP1）。

【0028】

次に、図3に示す第1ステージの処理サブルーチンに入る（STEP2）。第1ステージの処理では、ユーザによる回路記述ファイル21を開き（STEP3）、ファイルが終わりか否かを判定する（STEP4）。ファイルが終わりでないとは判定されると、拡張記述に関する記述か否かが判定される（STEP5）。ここで拡張記述であると判定されると拡張記述解析サブルーチンへジャンプし（STEP6）、拡張記述でないと判定されると、何も変換せずに一時バッファtempにタグ「A」を付加して格納し（STEP7）、STEP4の判定動作に戻る。そして、ファイルが終わりとは判定されるまで上述したSTEP4乃至STEP7の動作を繰り返す。STEP4でファイルが終わりとは判定されると、サブ

ーチンの呼び出し元（STEP 2）へ戻る。

【0029】

拡張記述サブルーチンでは、図4に示すように、フリップフロップに関する拡張記述か否かが判定される（STEP 8）。そして、フリップフロップに関する拡張記述であると判定されると、拡張回路記述の文法解析により情報を構造体FFの各変数へ格納し（STEP 9）、フリップフロップに関する拡張記述でないと判定されると、エラー処理を行ってプログラムを終了する（STEP 10）。上記構造体FFの各変数への情報の格納は、変数名Nameにフリップフロップの名前、BitNumにフリップフロップのビット数、ClusterNoにクラスタ番号、CtrlNamePtr*にフリップフロップの制御信号群の名前へのポインタ、ValPtr*にフリップフロップの制御信号群がそれぞれ1になった場合に代入すべき値または信号群へのポインタの情報をそれぞれ格納する。

【0030】

次に、構造体FFの実体へのポインタに、タグ「B」を付加して一時バッファtmpに格納する（STEP 11）。

【0031】

その後、サブルーチンの呼び出し元（STEP 5）へ戻る。

【0032】

上述した第1ステージの処理サブルーチンが終了すると、第2ステージの処理サブルーチンへジャンプする（STEP 12）。第2ステージの処理サブルーチンでは、まず図5に示すように、一時バッファtmpの終わりか否かが判定される（STEP 13）、終わりでないと判定されると一時バッファtmpからタグ及び格納情報を取得する（STEP 14）。そして、タグが「B」か否かが判定される（STEP 15）。タグが「B」でない場合には、格納情報をそのまま出力ファイルOUTに出力（STEP 16）してSTEP 13に戻って同様な処理を繰り返す。そして、上記STEP 13で一時バッファtmpの終わりと判定されると、サブルーチンの呼び出し元（STEP 12）へ戻る。一方、タグが「B」であった場合には、タグBの処理サブルーチンへジャンプする（STEP 17）

【0033】

タグBの処理サブルーチンは、図6に示すように、配列CTRLにゲートッドクロック機能を有効にする指定が存在し、タグBとともに格納された構造体FFのデータにおける変数ClusterNo(=CurrentNo)と別のClusterNo(=AppendNo)とのクラスタ結合指定が存在するか否かが判定される(STEP18)。クラスタ結合指定が存在する場合には、クラスタ結合サブルーチンへジャンプし(STEP19)、存在しない場合には配列CTRLにゲートッドクロック用セルを直接配置する指定が存在するか否かが判定される(STEP20)。指定が存在する場合には、フリップフロップ用のクロック信号として、配列CTRLに格納されているクロック信号をゲートッドクロック用に制御する回路を通過した後の信号名を使用するように変更する。また、制御する回路用のVerilog記述を出力ファイルOUTに出力する(STEP21)。一方、指定が存在しない場合には、フリップフロップ用のクロック信号として配列CTRLに格納されているクロック信号を使用することにする(STEP22)。

【0034】

上記STEP21及びSTEP22による処理結果に対して、配列CTRLにフリップフロップを非同期リセットに設定する項目が存在するか否かが判定される(STEP23)。存在する場合には、構造体FFの変数CtrlNamePtrで示される配列に配列CTRLに格納された「リセット信号」と同じものが存在した場合には、その信号を非同期リセット信号と見なして構造体FFの内容を、非同期リセットを有するフリップフロップとなるようなVerilog記述を生成し、出力ファイルOUTへ出力する(STEP24)。これに対し、存在しない場合には、構造体FFの変数CtrlNamePtrで示される配列に配列CTRLに格納された「リセット信号」と同じものが存在した場合には、その信号を同期リセット信号と見なして構造体FFの内容を、同期リセットを有するフリップフロップとなるようなVerilog記述を生成し、出力ファイルOUTへ出力する(STEP25)。その後、サブルーチンの呼び出し元(STEP

15)に戻る。

【0035】

上記STEP19において、クラスタ結合サブルーチンへジャンプした場合には、図7に示すように、結合対象のAppendNoが存在するか否かが判定され(STEP26)、存在する場合には、CurrentNoの構造体FFとAppendNoの構造体を結合して、1つのゲテッドクラスタを形成するようにCurrentNoの構造体FFの構造を変更する(STEP27)。次に、結合に必要な回路のうち、構造体FFだけで表現できないVerilog記述に関する記述を生成して出力ファイルOUTに出力する(STEP28)。その後、Verilog記述に関する情報を一時バッファtmpから削除する(STEP29)。そして、STEP26に戻って結合対象のAppendNoが存在しなくなるまでSTEP27乃至STEP29の処理を繰り返し、存在しなくなるとゲテッドクロック化の対象となったフリップフロップのレジスタ名などの情報を論理合成制御ファイルの形式で出力する(STEP30)。次に、サブルーチンの呼び出し元(STEP18)に戻る。

【0036】

図8は、上記図2乃至図7に示したプリプロセッサによるフリップフロップのリセットの同期／非同期指定を変更する場合の入力ファイルと出力ファイルを示している。ここでは、フリップフロップのリセット方式として同期リセット／非同期リセットを指定した場合の、それぞれの変換プログラムA(Program A)の出力結果を示す。

【0037】

図8において、“</”と“/>”はマーカー(marker)であり、“</”と“/>”で囲まれた部分の記述が変換対象となる。clk clockは変換後のハードウェア記述におけるクロック信号の名前(ここではclock)を指定するものである。delayは変換後のハードウェア記述におけるシミュレーションのディレイ値(ここでは2)を指定するものである。

【0038】

また、programAは、プリプロセッサ23の変換プログラムAを意味し

ており、option-SYNCは、上記programAに対して同期リセット方式でフリップフロップを初期化するハードウェア記述を出力することを指定し、option-ASYNCは、上記programAに対して非同期リセット方式でフリップフロップを初期化するハードウェア記述を出力することを指定する。

【0039】

上記のようなプリプロセッサへの入力ファイルに対して、フリップフロップのリセットの同期／非同期指定に応じた変換プログラムAによる処理結果が出力される。

【0040】

このように、プリプロセッサ23により、フリップフロップの同期／非同期指定を自由に変更可能な回路記述ファイルを生成できる。また、フリップフロップのリセットに関して「同期リセット」とした記述と「非同期リセット」とした記述の2つは、変換プログラムAを2回実行すれば得られる。これにより人手で作成した場合に比べて、短期間で「同期リセット」とした記述と「非同期リセット」の2つを得ることが可能である。

【0041】

図9は、ゲートドクロック機能をグルーピング指定なしで行った場合（クラスタ結合機能を使わない場合）のプリプロセッサ（変換プログラムA）23の入力ファイルと出力ファイルの例である。また、図10は、上記図9に示した出力ファイルから形成された回路例を示している。ここで、a00, a01はクロック信号を停止（ゲーティング）するための回路、b00, b02はラッチ、b01, b03は論理積回路、c00, c01, c02はフリップフロップ、d00, d01はクラスタを表している。

【0042】

プリプロセッサ23にゲートドクロック機能を指示する信号enable-gate-clockが入力されると、入力ファイルは変換プログラムAにより処理されて出力ファイルが生成される。

【0043】

図11は、ゲートドクロック機能をグルーピング指定して行った場合（クラスタ結合機能を使った場合）のプリプロセッサ（変換プログラムA）の入力ファイルと出力ファイルの例である。また、図12は、上記図11に示した出力ファイルから形成された回路例を示している。プリプロセッサ23には、ゲートドクロック機能を指示する信号 `enable gate-clock` とクラスタ結合機能を指示する信号 `grouping {FF00, FF01}` が入力される。クラスタ結合機能は変換プログラムAに対して `grouping {FF00, FF01}` などを指定することで機能する。ここで、`a00` はクロック信号を停止（ゲーティング）するための回路、`b00` はラッチ、`b01` は論理積回路、`c00`, `c01`, `c02` はフリップフロップ、`d00` はクラスタである。

【0044】

このように、フリップフロップの同期／非同期指定だけでなく、ゲートドクロックのクラスタ結合も変更可能な回路記述ファイルを生成できる。しかも、低消費電力化のためのゲートドクロック化において、より大きなクラスタをクラスタ結合機能により簡単に実現することができるために、クラスタサイズの最適値を求める作業などに要する時間を大幅に短縮できる。

【0045】

上述したように、この発明の実施の形態に係るプリプロセッサ、集積回路の設計システム及び集積回路の設計方法では、フリップフロップのハードウェア記述において、同期リセット付きフリップフロップであるか、非同期リセット付きフリップフロップであるかを特定しない形式で記述する。この際、上記フリップフロップに関する記述は、既存のハードウェア記述言語（第2のハードウェア記述言語）で記述されたファイル（第1の回路記述ファイル）の中に、第1のハードウェア記述言語で記述する。このフリップフロップに関する記述が埋め込まれた回路記述ファイルを変換プログラムA（Program A）で処理すると、フリップフロップに関する記述部は既存のハードウェア記述言語に変換されて出力される。このとき変換プログラムAに与える引数により、同期リセット方式のフリップフロップ記述を出力させることができる。同様に、非同期リセット方式のフリップフロップ記述を出力させることもできる。

【0046】

このように、ハードウェア記述言語の変換を変換プログラムAにより自動化することにより、「同期リセット」と「非同期リセット」のそれぞれの方式で記述されたハードウェア記述を簡単に得ることができる。また、例えば「同期リセット」方式の記述側を検証すれば、「非同期リセット」方式の記述の方も検証されたことになるので、2つのハードウェア記述の検証にかかる時間を半分にすることができる、これにより、LSIの開発等において開発期間を短縮することができる。

【0047】

また、フリップフロップのクロックをゲーテッドクロック化する指定を変換プログラムAに対して指定することが可能であり、この場合にはフリップフロップのクロックをゲーティングした記述が生成される。更に、フリップフロップの記述Xおよび記述Yが存在する場合に、記述Xと記述Yのクロックを共通化し、共通化されたクロックに対してゲーテッドクロック化を行うこともできる（クラスタ結合）。

【0048】

上述したように、様々なコンフィギュレーションに従ったRTL記述を容易に得ることが可能となる。

【0049】

従って、上記のような構成並びに設計方法によれば、フリップフロップの同期／非同期指定及びゲーテッドクロックのクラスタ結合を変更可能な回路記述ファイルをプリプロセッサ23で生成できるので、開発期間の短縮とコストの低減を図ることが可能となる。しかも、人手による作業ではバグの混入による修正が必要になり、修正のたびに機能検証作業が発生するが、自動化により検証作業が不要となり、この点からも開発期間の短縮やコストの低減が図れる。

【0050】

なお、上記実施の形態では、説明を分かりやすくするために、回路記述ファイル21とプリプロセッサ制御ファイル22が別のファイルの場合を例にとって説明したが、1つのファイルにまとめても良いのは勿論である。

【0051】

以上実施の形態を用いてこの発明の説明を行ったが、この発明は上記実施の形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。また、上記実施の形態には種々の段階の発明が含まれており、開示される複数の構成要件の適宜な組み合わせにより種々の発明が抽出され得る。例えば実施の形態に示される全構成要件からいくつかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題の少なくとも1つが解決でき、発明の効果の欄で述べられている効果の少なくとも1つが得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【0052】

【発明の効果】

以上説明したように、この発明によれば、フリップフロップの同期／非同期指定及びゲートドクロックのクラスタ結合を変更可能な回路記述ファイルを生成できるプリプロセッサが得られる。

【0053】

また、開発期間を短縮し、コストを低減できる集積回路の設計システムとこのシステムによる集積回路の設計方法が得られる。

【図面の簡単な説明】

【図1】 この発明の実施の形態に係るプリプロセッサ及び集積回路の設計システムについて説明するためのもので、システム全体の概略構成を示す機能ブロック図。

【図2】 図1に示したシステムにおけるプリプロセッサの概略的な動作を示すフローチャート。

【図3】 図1に示したシステムにおけるプリプロセッサの動作を示すもので、第1ステージの処理サブルーチンを示すフローチャート。

【図4】 図1に示したシステムにおけるプリプロセッサの動作を示すもので、拡張記述解析サブルーチンを示すフローチャート。

【図5】 図1に示したシステムにおけるプリプロセッサの動作を示すもので、第2ステージの処理サブルーチンを示すフローチャート。

【図6】 図1に示したシステムにおけるプリプロセッサの動作を示すもので、タグBの処理サブルーチンを示すフローチャート。

【図7】 図1に示したシステムにおけるプリプロセッサの動作を示すもので、クラスタ結合サブルーチンを示すフローチャート。

【図8】 図2乃至図7に示したプリプロセッサによるフリップフロップのリセットの同期／非同期指定を変更する場合の入力と出力の関係について説明するための図。

【図9】 図2乃至図7に示したプリプロセッサによるクラスタ結合機能を使用しない場合の入力ファイルと出力ファイルとの関係について説明するための図。

【図10】 図1に示した集積回路の設計システムを用い、図2乃至図7に示したプリプロセッサによるクラスタ結合機能を使用しないで形成した回路の構成例を示す回路図。

【図11】 図2乃至図7に示したプリプロセッサによるクラスタ結合機能を使用した場合の入力ファイルと出力ファイルとの関係について説明するための図。

【図12】 図1に示した回路システムを用い、図2乃至図7に示したプリプロセッサによるクラスタ結合機能を使用して形成した回路の構成例を示す回路図。

【図13】 従来の集積回路の設計システム及び集積回路の設計方法について説明するための機能ブロック図。

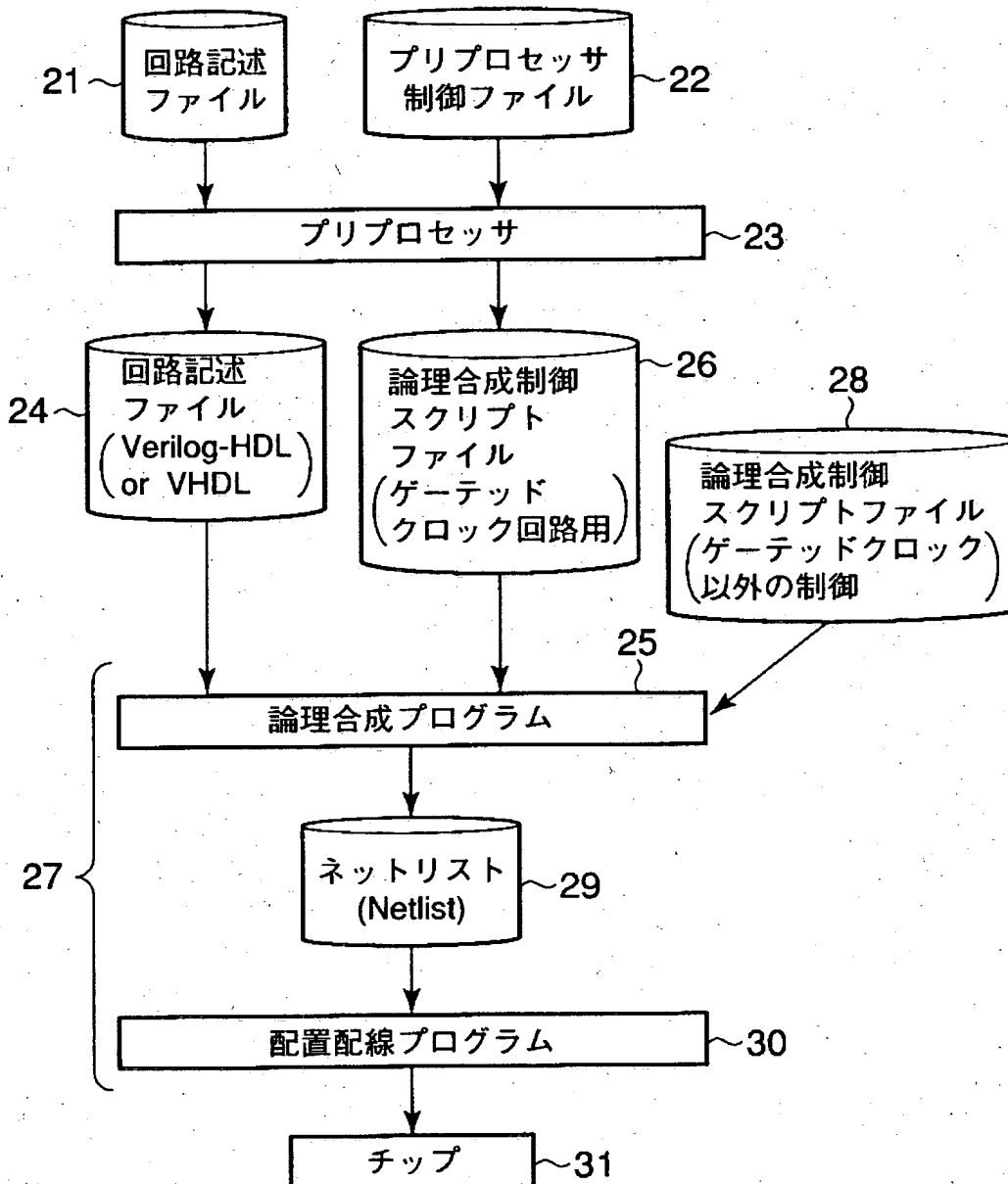
【符号の説明】

21…回路記述ファイル（第1の回路記述ファイル）、22…プリプロセッサ制御ファイル、23…プリプロセッサ、24…回路記述ファイル（第2の回路記述ファイル）、25…論理合成プログラム、26…論理合成制御スクリプトファイル、27…論理合成ツール（コンピュータ）、28…論理合成スクリプトファイル、29…ネットリスト、30…配置配線プログラム、31…チップ。

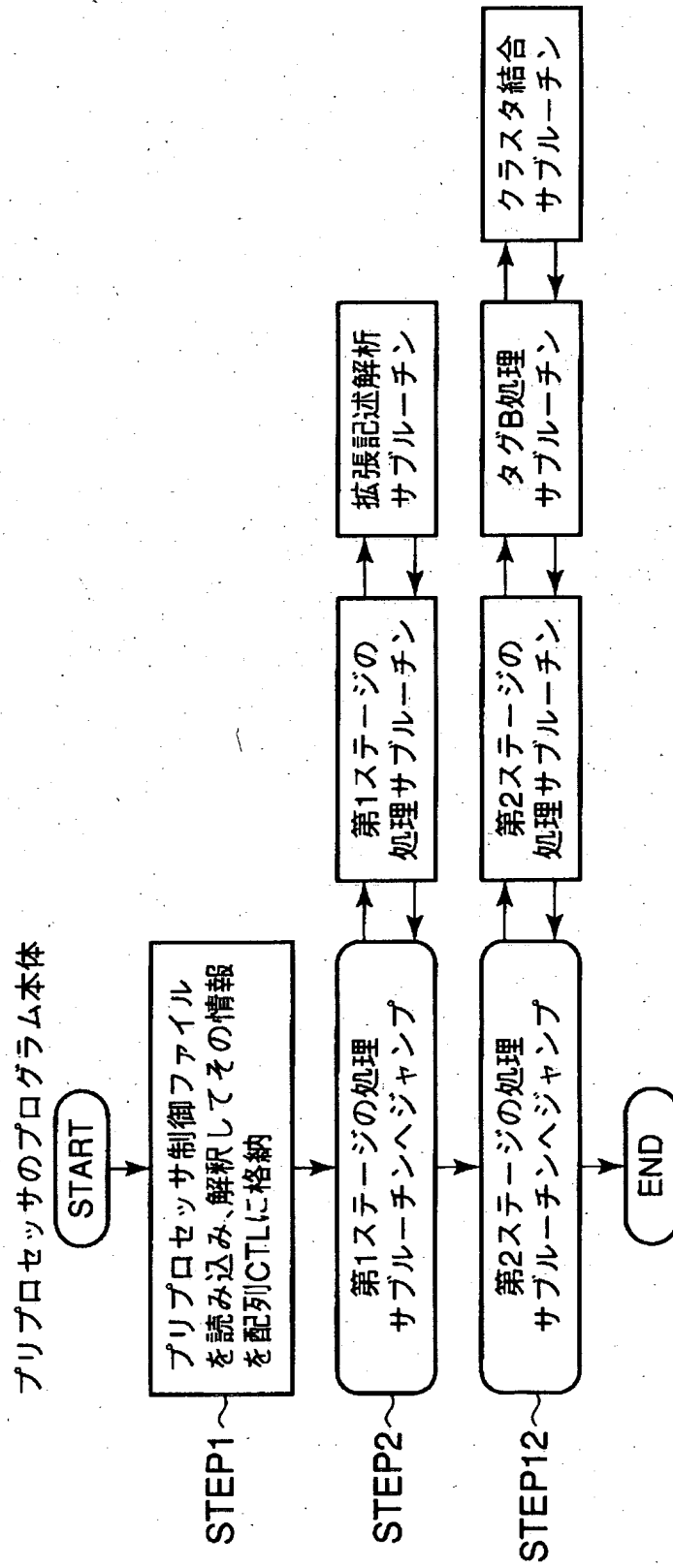
【書類名】

図面

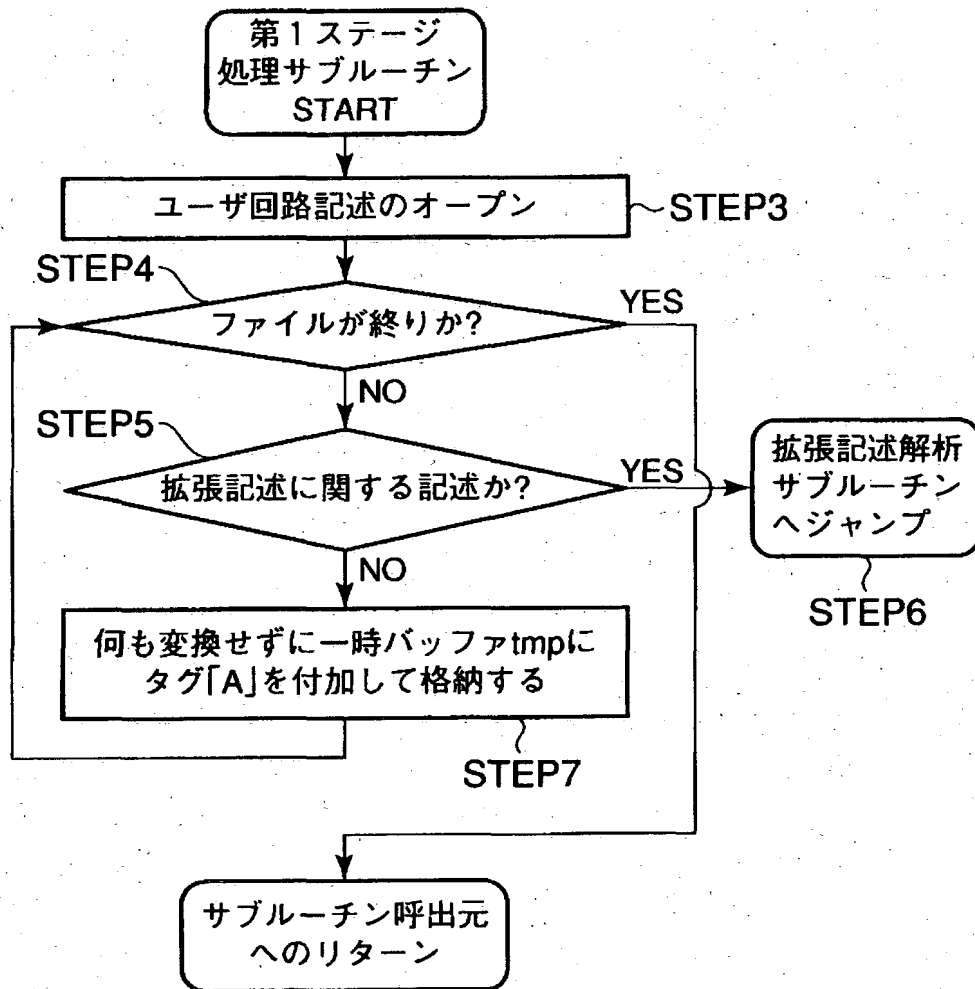
【図 1】



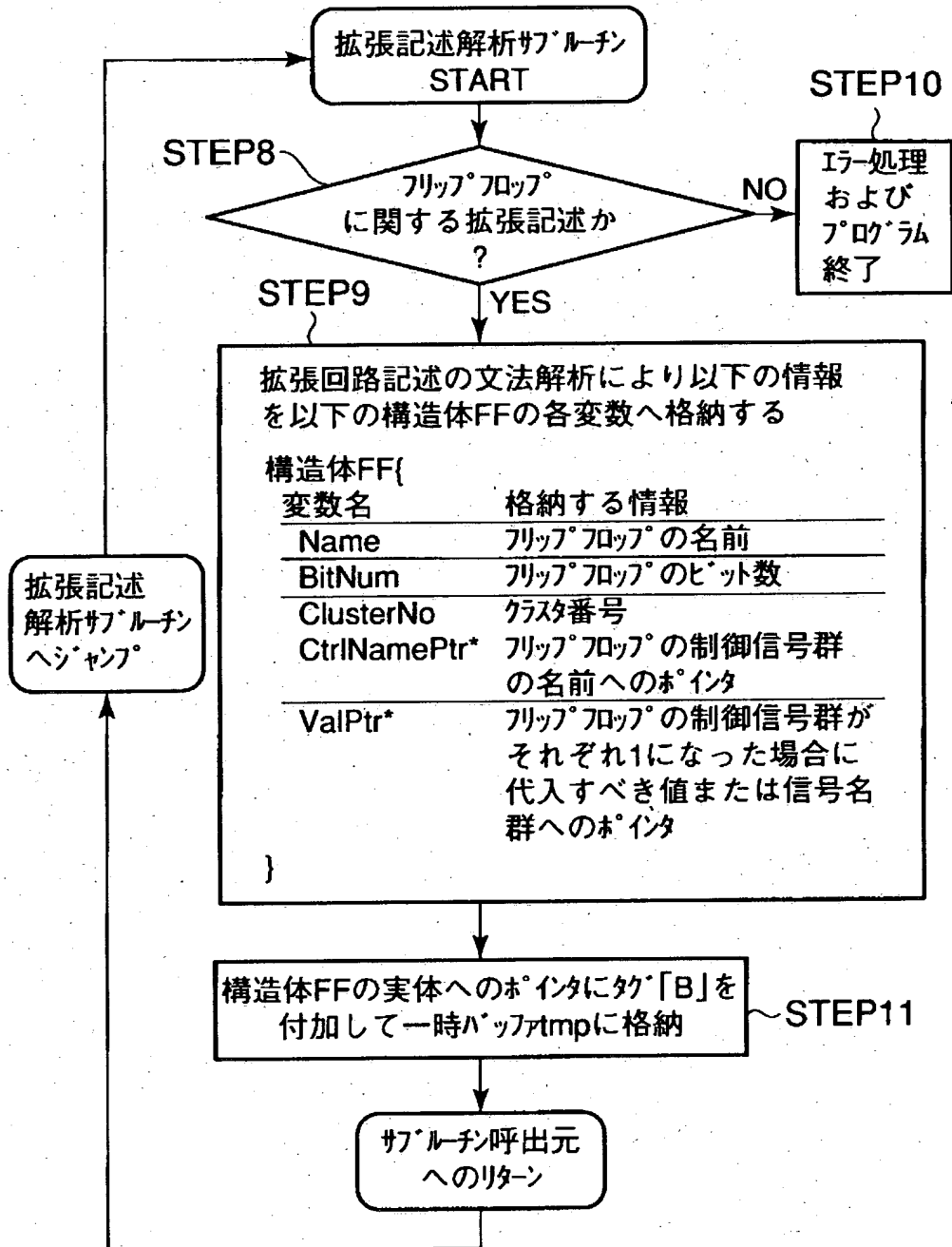
【図2】



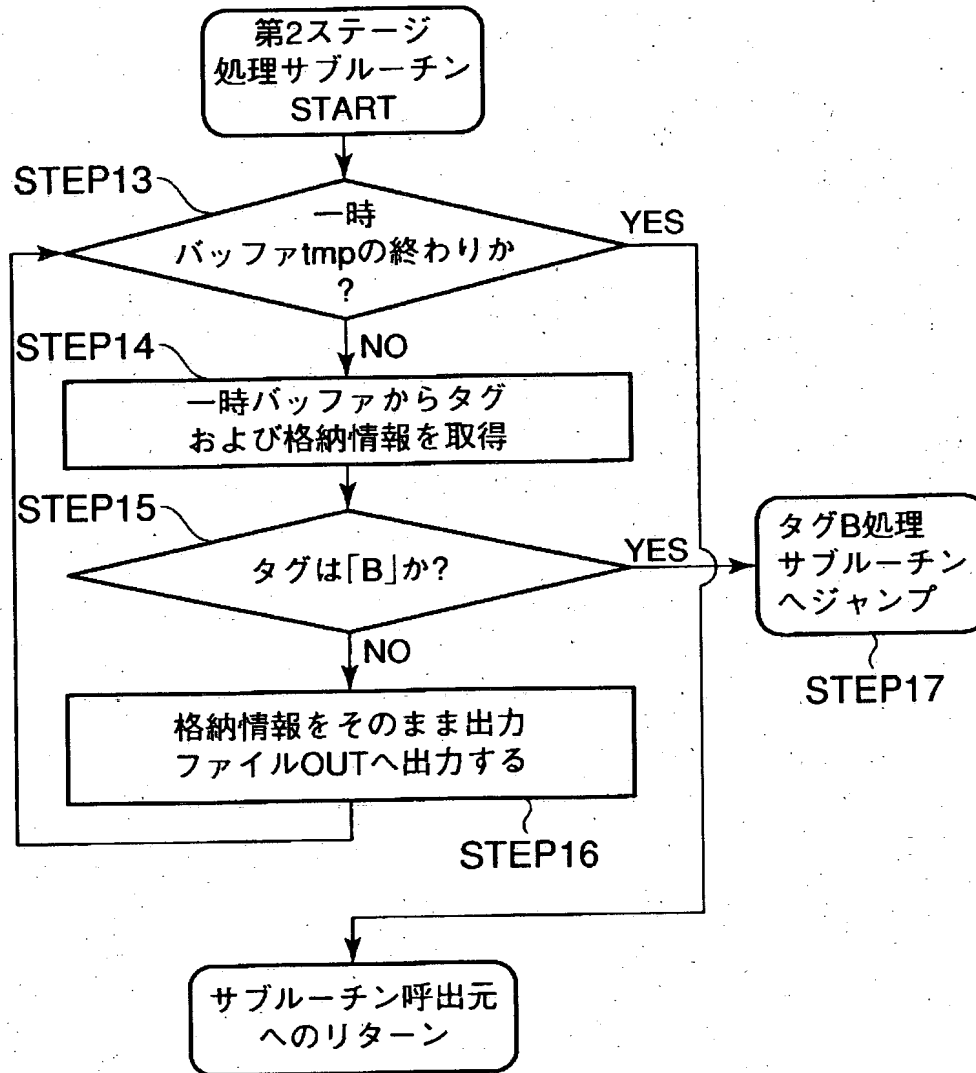
【図3】



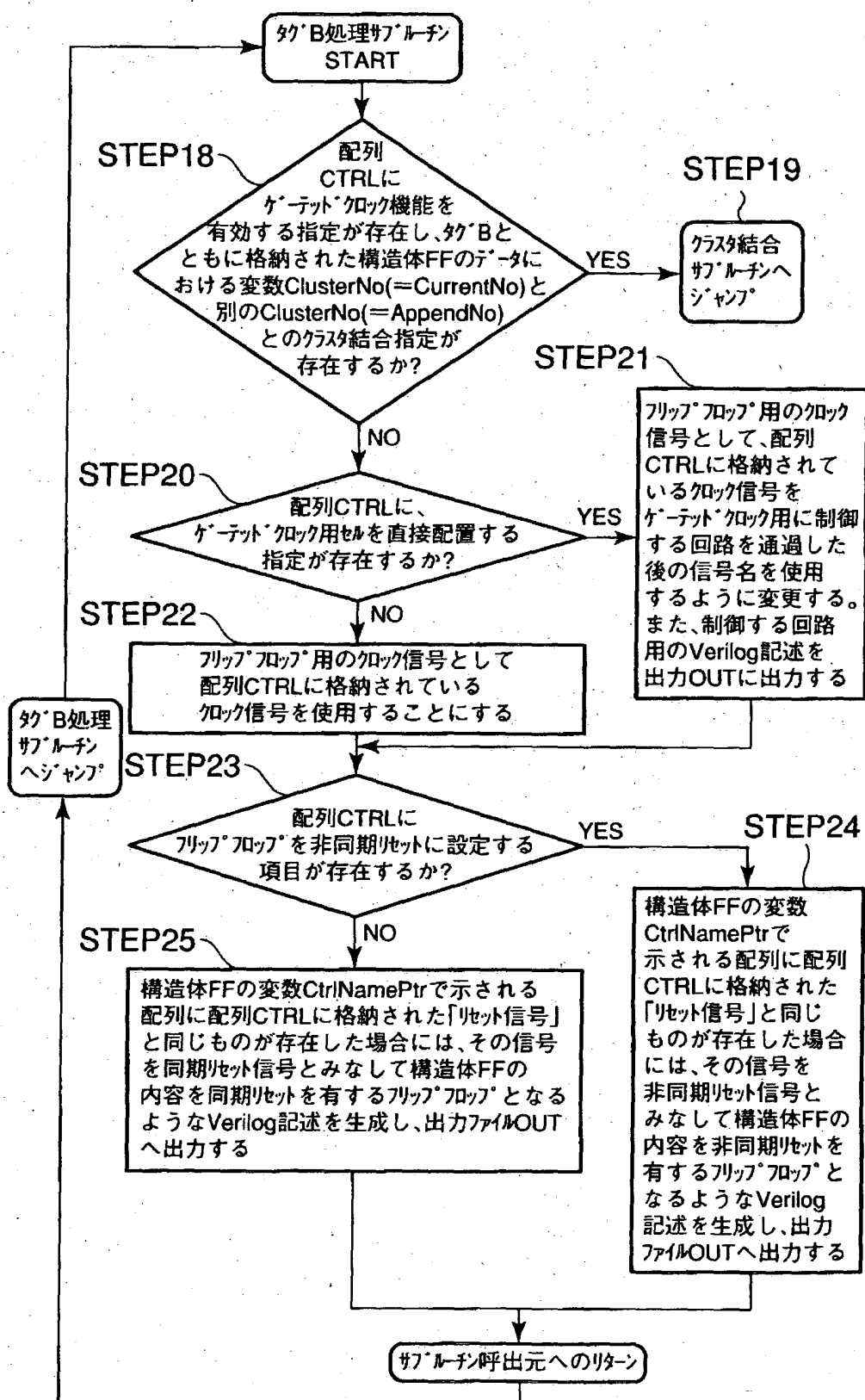
【図4】



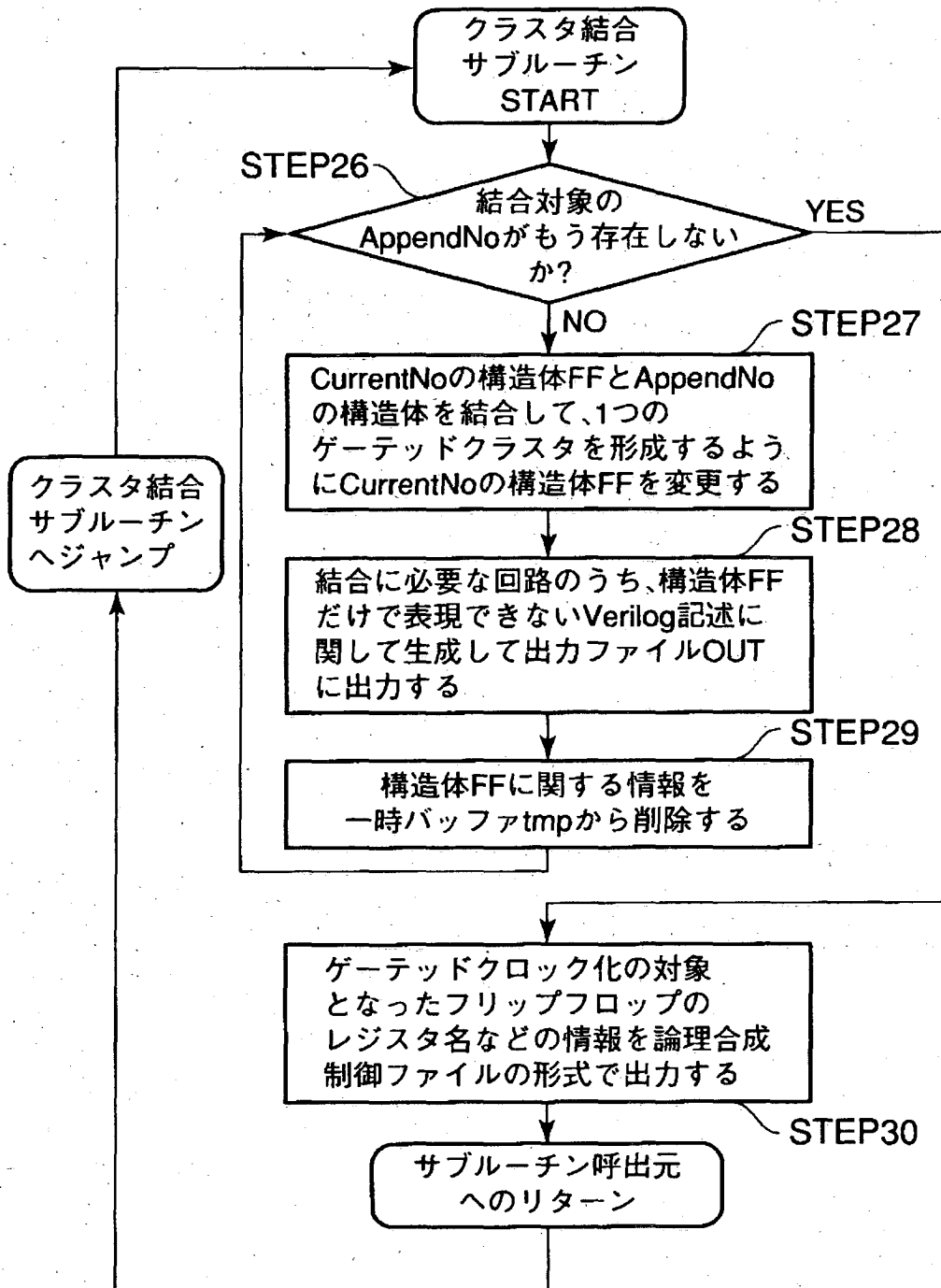
【図5】



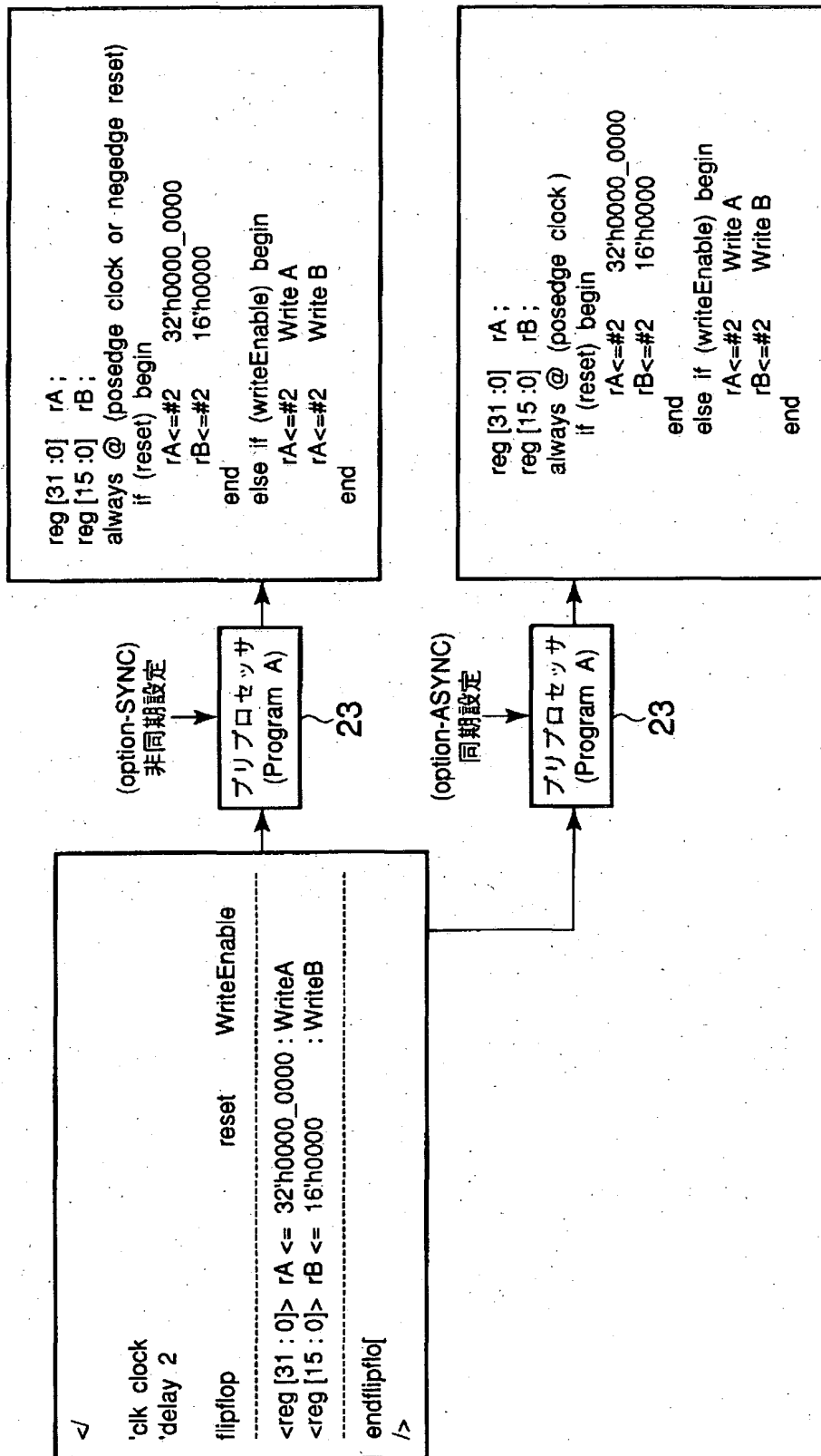
【図6】



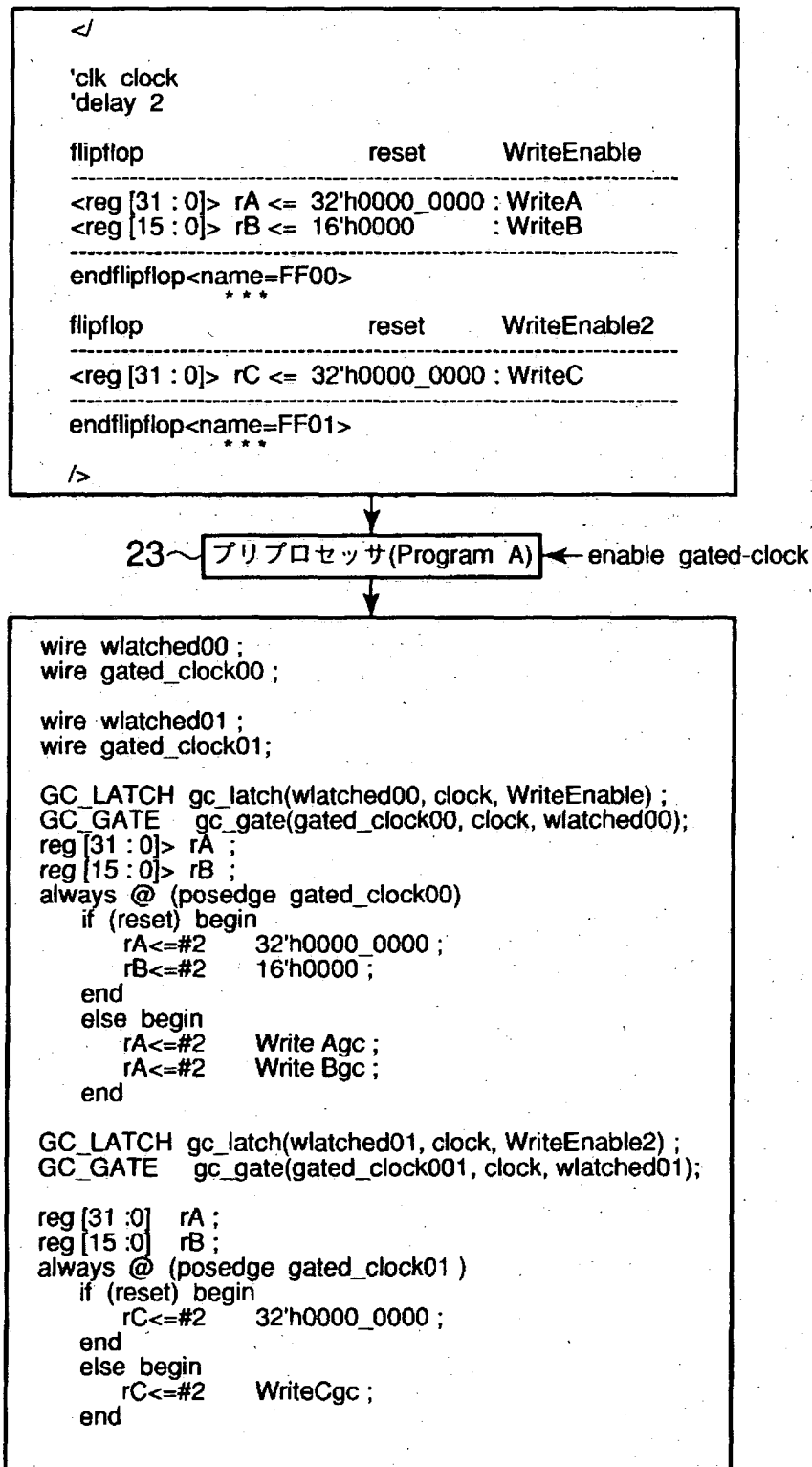
【図7】



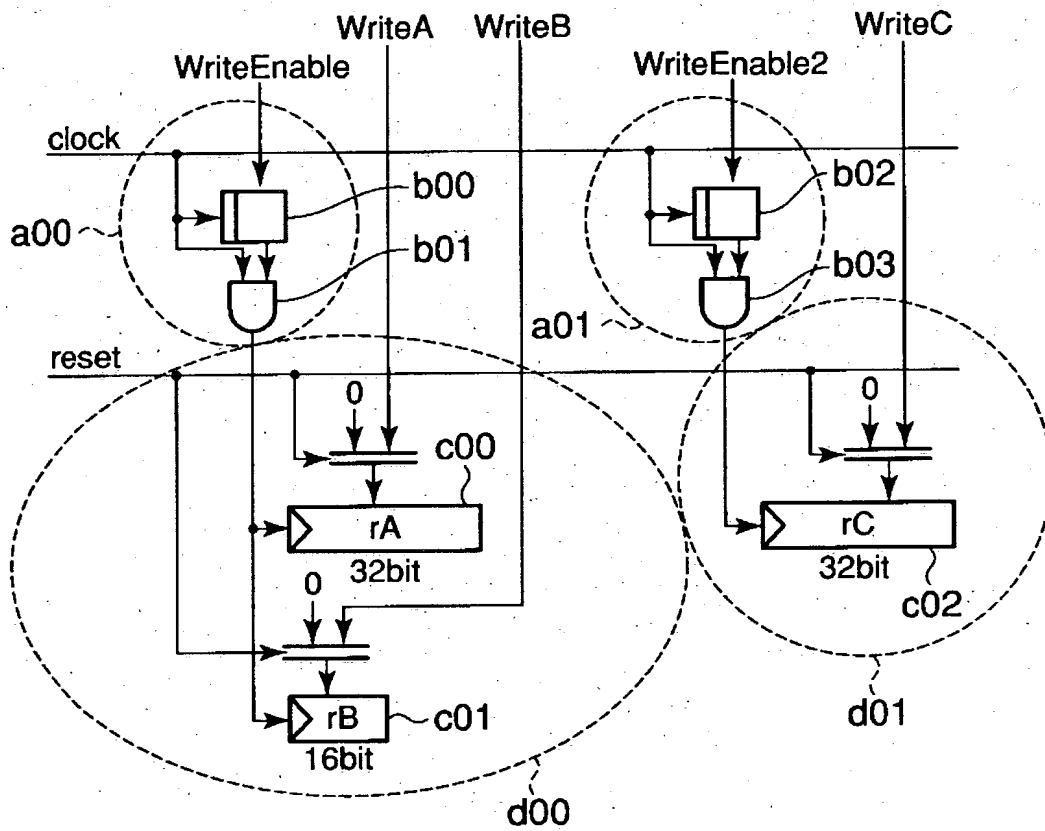
【図 8】



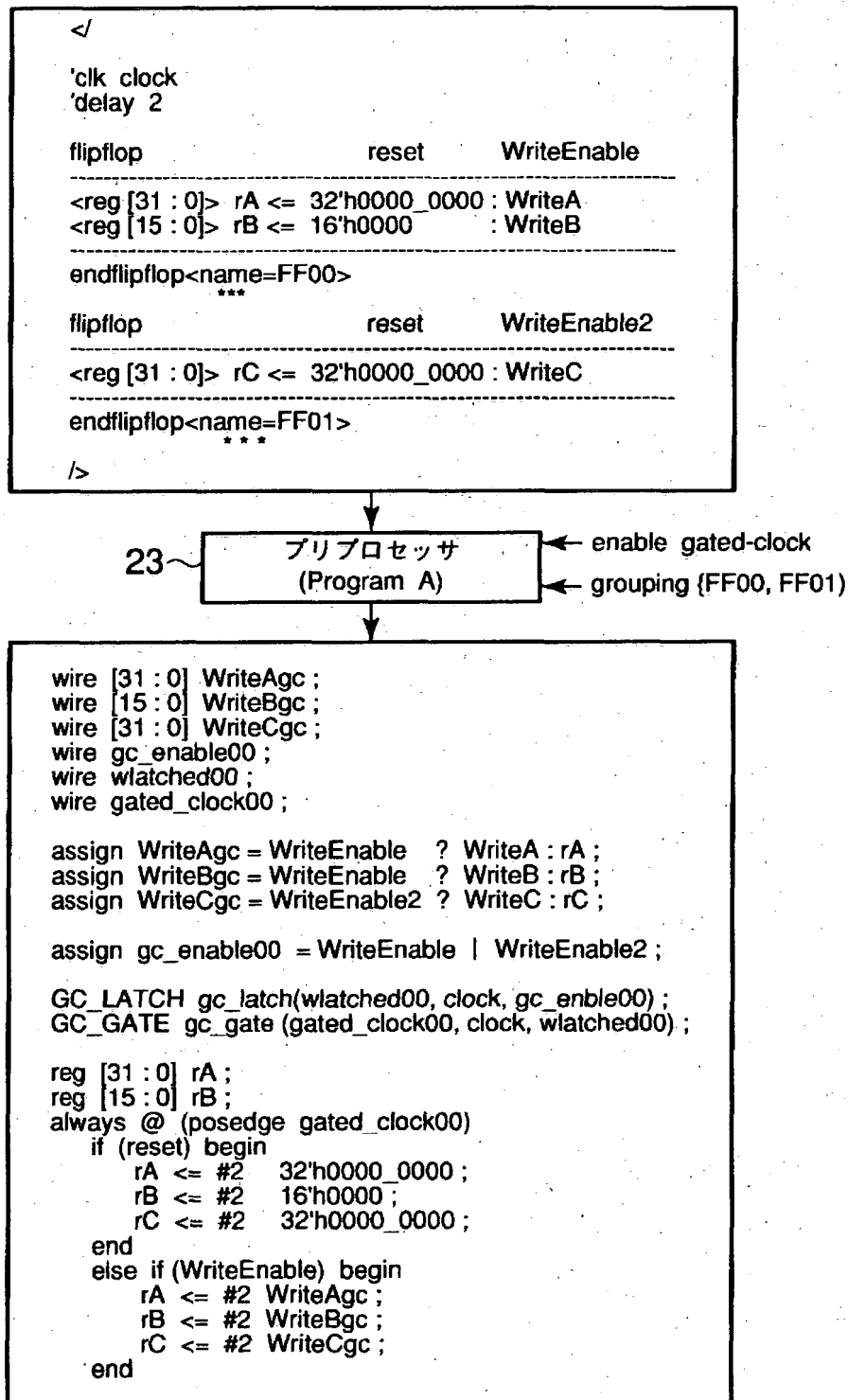
【図9】



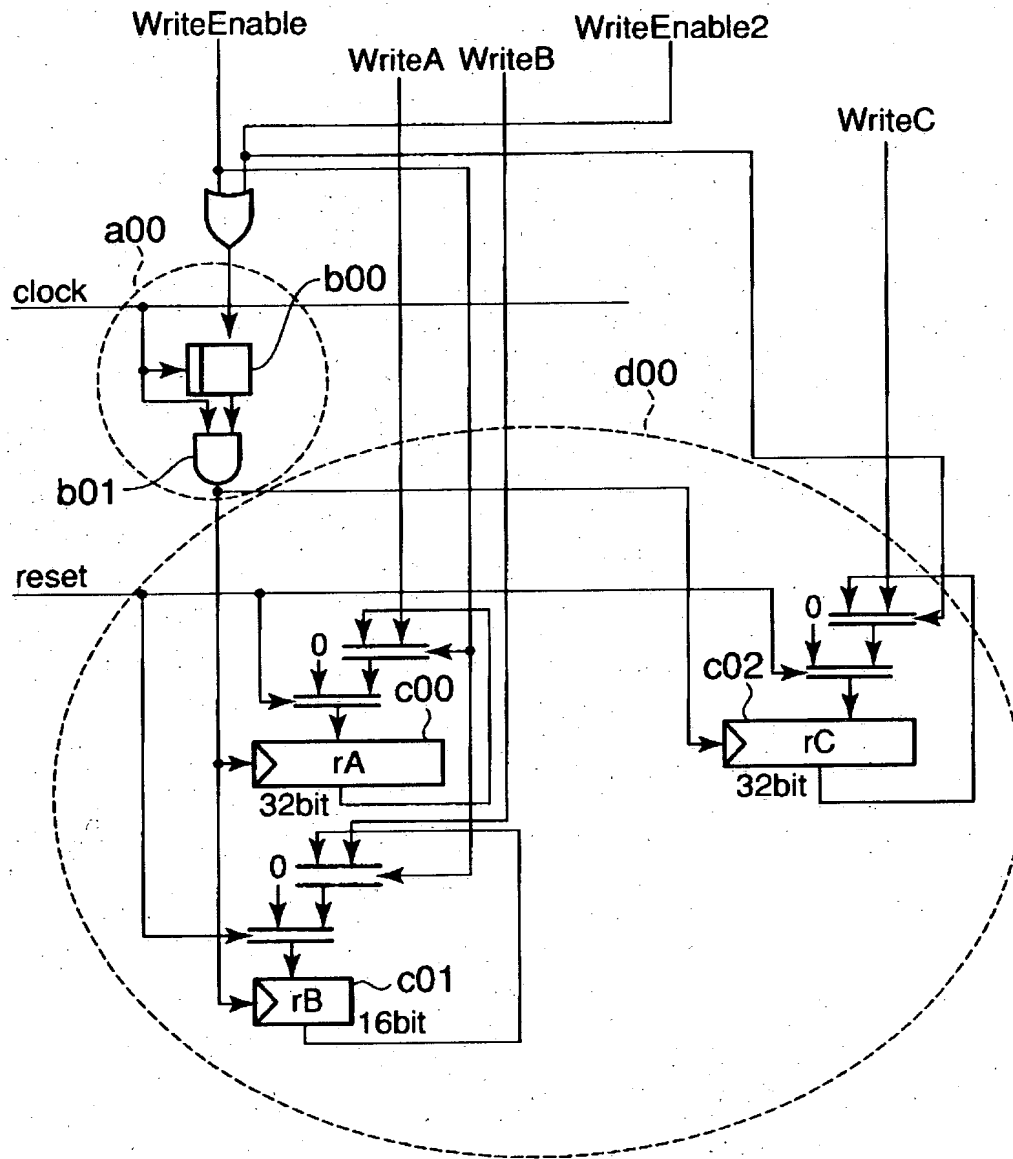
【図10】



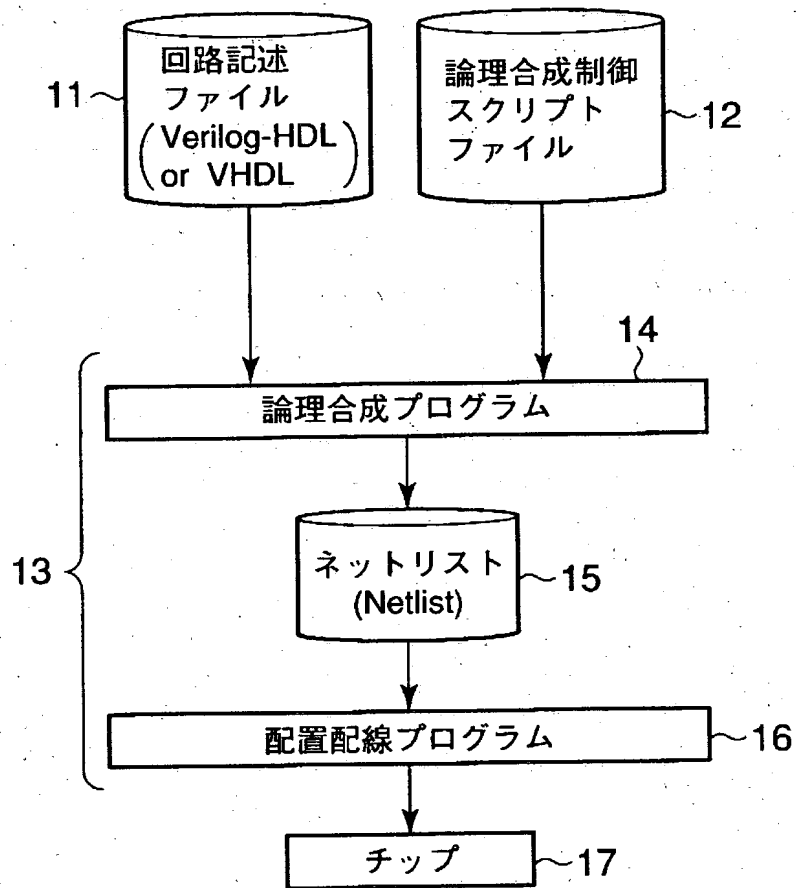
【図 11】



【図 12】



【図13】



【書類名】 要約書

【要約】

【課題】 フリップフロップの同期／非同期指定及びゲートドクロックのクラスタ結合を変更可能な回路記述ファイルを生成できるプリプロセッサを提供することを目的としている。

【解決手段】 プリプロセッサ 23 は、ユーザによって記述されたプリプロセッサ制御ファイル 22 に基づいて、ユーザによって記述され、第 1 のハードウェア記述言語と第 2 のハードウェア記述言語が混在する第 1 の回路記述ファイル 21 の処理を行い、前記第 1 の回路記述ファイル 21 における第 1 のハードウェア記述言語で記述された少なくとも一部を第 2 のハードウェア記述言語に変換して第 2 の回路記述ファイル 24 を生成し、出力することを特徴としている。これによって、人手を介在することなく、フリップフロップの同期／非同期指定及びゲートドクロックのクラスタ結合を変更可能な回路記述ファイルを生成できる。

【選択図】 図 1

特2003-096687

出 願 人 履 歷 情 報

識別番号

[000003078]

1. 変更年月日 2001年 7月 2日

[変更理由] 住所変更

住 所 東京都港区芝浦一丁目1番1号

氏 名 株式会社東芝